An Efficient Global Motion Estimation based on Robust Estimator

Jaehwan Joo, Yoonsik Choe

School of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
E-mail: jhjoo80@yonsei.ac.kr, yschoe@yonsei.ac.kr

ABSTRACT

In this paper, a new efficient algorithm for global motion estimation is proposed. This algorithm uses a previous 4-parameter model based global motion estimation algorithm and M-estimator for improving the accuracy and robustness of the estimate. The first algorithm uses the block based motion vector fields and which generates a coarse global motion parameters. And second algorithm is M-estimator technique for getting precise global motion parameters. This technique does not increase the computational complexity significantly, while providing good results in terms of estimation accuracy. In this work, an initial estimation for the global motion parameters is obtained using simple 4-parameter global motion estimation approach. The parameters are then refined using M-estimator technique. This combined algorithm shows significant reduction in mean compensation error and shows performance improvement over simple 4-parameter global motion estimation approach.

Keywords: global motion estimation, 4-parameter, M-estimator

1. Introduction

With the development of multi-media technology and the variety of application environments, the relation of quality and compression in video coding is getting more critical. Video standard such as MPEG-1, MPEG-2, MPEG-4, MPEG-7, H.261, H.263, and H.264 adopts advance techniques for achieving the high coding efficiency. Global motion estimation (GME) and compensation (GMC) is one of these advance techniques.

GME is an important tool widely used in computer vision, video processing, and other fields. Some examples are virtual reality, image registration, video segmentation, video processing in MPEG-7, statistic-sprite generations in MPEG-4 Sprite coding, and global motion compensation in MPEG-4 ASP. In this paper, we focus on the GME for video compression [1].

Conventional MC methods use translational motion models to describe the horizontal and vertical motion of objects. However, moving objects have more complex motion, such as rotation, pan, tilt, zoom, etc. To solve this problem, global motion estimation is proposed

There are many global motion estimation techniques which are well developed and can be applied for motion compensated frame prediction. Usually, the camera motions are modeled with a number of global motion parameters. For example, two to nine global motion parameter models are proposed for representing the associated camera motions.

The computational complexity and accuracy are main concerning of GME algorithm. GME's performance will increase when we use higher accuracy model (i.e. with more parameters), while computational complexity will also increase at the same time. So many algorithms have been proposed to give a trade off between complexity and accuracy.

Renen Coudray and Bernard Besserer [2][3] proposed easy and simple four-parameter global motion estimation method which is based on motion vector field for reducing computational complexity and fast estimation of global motion parameters. This method has robust to local motion. However as the estimate is generated from block based motion vector field, the accuracy is dependent on the motion vector. But it is not sufficiently accurate.

With respect to the above shortcomings, the proposed global motion estimation algorithm adopts robust statistics and maximum-likelihood-theory, a so called M-estimator for reducing the influence of outliers in every iteration. Using first easy and simple global motion estimation algorithm, we get coarse global motion parameters that are initial values at second step. And using M-estimator, we get more precise global motion parameters. Experimental results reveal that this combined global motion estimation algorithm has less bit rate and overall improvement in PSNR compared to previous global motion estimation algorithm.

This paper is organized as follows: In Section 2, previous simple four-parameter global motion estimation algorithm is introduced. In Section 3, proposed global motion estimation scheme is described and presents the experimental results. Finally, a conclusion is given in Section 4.

2. Global Motion Models and Previous Global Motion Estimation Algorithm

2.1 Global Motion Models

Various practical 2D parametric global motion models are summarized in Table 1. Different models are denoted as M_i with the number of parameters i as subscripts. As a general rule, higher order models have more complex motions.

Generally, GME's performance will increase when we use higher accuracy model (i.e. with more parameters), while computational complexity will also increase at the same time.

Table 1: Different Global Motion Models

Motion Model	Number of Parameter	Transform
Translation M_2	2	$x' = x + c$ $y' = y + f$
Geometric M_4	4	$x' = ax + by + c$ $y' = -bx + ay + f$
Affine M_6	6	$x' = ax + by + c$ $y' = dx + ey + f$
Perspective M_8	8	$x' = \frac{ax + by + c}{px + qy + 1}$ $y' = \frac{dx + ey + f}{px + qy + 1}$

2.2 Previous Global Motion Estimation Algorithm

In [2], simple 4-parameter global motion estimation which is based on motion vector field is proposed. The used notation is:

V_x, V_y : *motion vector components for each sample within the image*

x, y : *spatial position of the sample*

This algorithm attempts to estimate zoom, rotation and translation parameters form motion vector field. If we assume A is scaling factor and θ is rotation factor,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} A\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} x'-x \\ y'-y \end{bmatrix} = \begin{bmatrix} A\cos\theta - 1 & -\sin\theta \\ \sin\theta & \cos\theta - 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2)$$

Let $z = A\cos\theta - 1$ and $r = A\sin\theta$

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} z & -r \\ r & z \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (3)$$

$$V_T(x,y) = \begin{bmatrix} t_x \\ t_y \end{bmatrix}, V_z(x,y) = \begin{bmatrix} z \cdot x \\ z \cdot y \end{bmatrix}, V_r(x,y) = \begin{bmatrix} -r \cdot x \\ r \cdot y \end{bmatrix} \quad (4)$$

where z and r are the zoom and rotation factors and t_x and t_y are horizontal and vertical components of the translation respectively. It can be deduced that

$$\frac{\partial V_x}{\partial x} = \frac{\partial V_y}{\partial y} = z, \quad -\frac{\partial V_x}{\partial y} = \frac{\partial V_y}{\partial x} = r \qquad V_x = t_x, \quad V_y = t_y \quad (5)$$

A zoom and rotation will affect the estimation of the translation. So a zoom and rotation must be compensated before translation can be calculated. Maximum histogram bin on z and r are derived from the partial derivatives. Translation parameters are then estimated by compensating z and r parameters on motion vectors. And same method that is histogram bin derived from the partial derivatives is used for getting translation parameters. A flow chart of this algorithm is given in Fig.1.

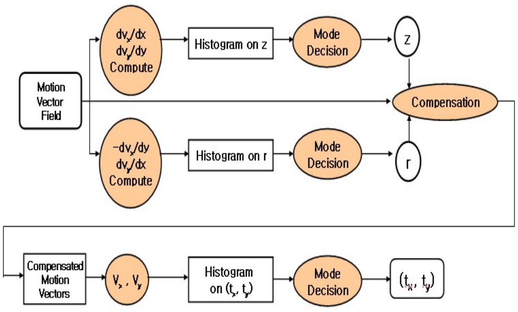

Fig.1: 4-Parameter Global Motion Estimation Algorithm

Before we extract global motion parameter using above algorithm, we do not use homogeneous region's motion vectors. Because homogeneous region's motion vector tends to be (0, 0), z and r parameters also tend to (0, 0). It is outlier when we extract global motion parameters. So using 'Variance' or 'DCT Coefficient', we reject homogeneous region's motion vectors.

But this algorithm is ineffective when objects are very large and have movement which is differing from camera motions. At this case, this algorithm decides that object's motion is global motion. See the below Fig. 2.

Fig. 2: "Mobile" CIF (Frame Number: #0, #50, #100)

This sequence has large object that is 'calendar'. The object move up and down continuously. But camera has only translation (right to left side) and zoom motions. This four-parameter global motion estimation algorithm decides that up and down translation motion is global motion. That is incorrect.

3. Proposed Global Motion Estimation Algorithm

3.1 M-estimator Algorithm

M-estimator is generalizations of the usual maximum likelihood estimates (Huber, 1981; Ray, 1983; Hampel et al., 1986). Classically a parameter V is obtained by maximizing the likelihood function L, i.e. if x_i is the residual of ith data point, optimal parameter V^* is given by

$$V^* = \arg\max(L = \prod_i f(x_i | V)) \quad (6)$$

The estimators of type M are solutions of the more general structure

$$V^* = \arg\min(M = \sum \rho(x_i, V)) \quad (7)$$

Where the function $\rho(\cdot)$ is a symmetric positive definite function with a unique minimum at zero and is chosen to be increasing slower than quadratically.

Instead of solving this problem directly, we can reformulate it as an iterated weighed least-square problem. This is, for estimating a parameter vector $V=[v_1,v_2,...,v_n]^T$, the M-estimator of V based on the function $\rho(x_i)$, is the solution of following n-equations:

$$\sum\psi(x_i,V)\frac{\partial x_i}{\partial j}=0,\quad j=1,2,..,n \tag{8}$$

where the derivative $\psi=d\rho(x)/dx$ is called *the influence functions*. Several M-estimators' influence functions are summarized in Table 2.

Table 2: Robust Functions

Type	$\psi(x)$	$\rho(x)$	**Range of** x
Tukey	$\begin{cases}(1-x^2)^2\\0\end{cases}$	$\begin{cases}\frac{1}{6}[1-(1-x^2)^3]\\\frac{1}{6}\end{cases}$	$\|x\|\le 1$ $\|x\|>1$
Huber	$\begin{cases}1\\\tau\frac{\operatorname{sgn}(x)}{x}\end{cases}$	$\begin{cases}x^2\\2k\|x\|-x^2\end{cases}$	$\|x\|\le k$ $\|x\|>k$
Andrews	$\begin{cases}\frac{\sin(\pi x)}{\pi x}\\x\end{cases}$	$\begin{cases}\frac{1}{\pi^2}[1-\cos(\pi x)]\\\frac{1}{\pi^2}\end{cases}$	$\|x\|\le 1$ $\|x\|>1$

We apply four-parameter model. It described the motion of a point (x,y) to its transformed position (x',y') and can express translation, rotation, and scaling

$$\begin{aligned}x'&=(z+1)x-ry+t_x=z'x-ry+t_x\\y'&=rx+(z+1)y+t_y=rx+z'y+t_y\end{aligned} \tag{9}$$

The motion parameters combined in parameter vector.

$$\Xi=(z',r,t_x,t_y)^T \tag{10}$$

We use the center coordinated of the corresponding block for $(x^{(n)},y^{(n)})$. Such an equation can be formulated for all N motion vectors and all these equations can be combined in a single linear matrix equation.

$$V=H\cdot\Xi \tag{11}$$

with

$$V=\begin{bmatrix}v_x^{(1)}+x^{(1)}\\v_y^{(1)}+y^{(1)}\\.\\.\\v_x^{(N)}+x^{(N)}\\v_y^{(N)}+y^{(N)}\end{bmatrix},\quad H=\begin{bmatrix}x^{(1)}&-y^{(1)}&1&0\\y^{(1)}&x^{(1)}&0&1\\.&.&.&.\\.&.&.&.\\x^{(N)}&-y^{(N)}&1&0\\y^{(N)}&x^{(N)}&0&1\end{bmatrix} \tag{12}$$

The solution of equation (8) is given by

$$\Xi=(H^T\cdot H)^{-1}\cdot H^T\cdot V \tag{13}$$

Let $\Xi(k)$ be the solution of the *k*-th iteration. From these parameters, we can compute:

$$\delta^{(n)}=f(\Xi(k))=\begin{bmatrix}z'(k)x^{(n)}-r(k)y^{(n)}+t_x(k)-x^{(n)}\\r(k)x^{(n)}+z'(k)y^{(n)}+t_y(k)-y^{(n)}\end{bmatrix} \tag{14}$$

We define a measure for accuracy of the of $\Xi(k)$

$$\varepsilon^{(n)}=\sqrt{(v_x^{(n)}-\delta_x^{(n)})^2+(v_y^{(n)}-\delta_y^{(n)})^2} \tag{15}$$

$$\mu_\varepsilon=\frac{1}{N}\sum_n\varepsilon^{(n)} \tag{16}$$

We use Tukey's Biweight function.

$$w(\varepsilon)=\begin{cases}(1-(\frac{\varepsilon}{c\mu_\varepsilon})^2) & \varepsilon\le c\mu_\varepsilon\\0 & \varepsilon>c\mu_\varepsilon\end{cases} \tag{17}$$

where c is tuning constant that is used to adjust the sensibility of the algorithm.

Finally, the robust estimate is calculated in every iteration as

$$\Xi=(H^T\cdot W\cdot H)^{-1}\cdot H^T\cdot W\cdot V \tag{18}$$

with the weighting matrix

$$W=\begin{bmatrix}w^{(1)}&0&.&.&0\\0&w^{(1)}&.&.&.\\.&.&.&.&.\\.&.&.&w^{(N)}&0\\0&.&.&0&w^{(N)}\end{bmatrix} \tag{19}$$

3.2 Coding of Global Motion Parameters

'*z*' and '*r*' are encoded directly using the variable length codes listed in Table 3~5. And fixed codeword of 6 bits at each 't_x' and 't_y' is allocated for translation parameters.

Table 3: '*z*' Parameter $(z=A\cdot\cos\theta-1)$

θ \ **A**	**0.9**	**0.94**	**1.0**	**1.06**	**1.1**
-0.04 π	-0.1070	-0.0674	-0.0078	0.05164	0.09132
-0.02 π	-0.1017	-0.0618	-0.0019	0.05790	0.09782
0	0.1	-0.06	0	0.06	0.1
0.02 π	-0.1017	-0.0618	-0.0019	0.05790	0.09782
0.04 π	-0.1070	-0.0674	-0.0078	0.05164	0.09132

Table 4: '*r*' Parameter $(r=A\cdot\sin\theta)$

θ \ **A**	**0.9**	**0.94**	**1.0**	**1.06**	**1.1**
-0.04 π	-0.1128	-0.1178	-0.125	-0.1328	-0.1378
-0.02 π	-0.05651	-0.0590	-0.062	-0.0665	-0.0690
0	0	0	0	0	0
0.02 π	0.05651	0.05902	0.0627	0.06655	0.06906
0.04 π	0.11279	0.11781	0.1253	0.13285	0.13786

Table 5: VLC Table for ‘A’ & ‘θ’

A	Codeword	θ	Codeword
1.0	0	0	0
0.94	10	-0.02 π	10
1.06	110	0.02 π	110
0.9	1110	-0.04 π	1110
1.1	1111	0.04 π	1111

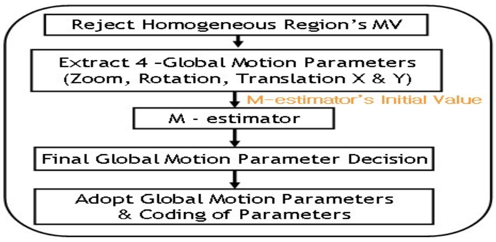

Fig. 3: Flowchart of Proposed Algorithm

3.3 Experimental Results and Analysis

The proposed algorithm was implemented on H.264/AVC JM 10.1 provided by JVT. The test conditions are as followed

Table 6: Test Conditions

H.264/AVC JM 10.1	Profile	Baseline
	Entropy Coding	CAVLC
	Quantization Parameter	20, 24, 28, 32, 36, 40
	Rate Control	Off
	RDO	Off
	ME Range	± 32
Test Sequence	‘Mobile’, CIF	Translation, Zoom
	‘Waterfall’, CIF	Zoom
	‘Dragon’, CIF	Zoom, Rotation

Our algorithm outperforms the standard H.264/AVC and previous algorithm in bit rate saving and PSNR improvement. This algorithm is more effective when we use low bit rate coding compared to high bit rate coding. Average bit rate saving is 1% and average PSNR improvement is 0.2 dB. At ‘Mobile’ sequence, performance gain is much better than previous global motion estimation algorithm.

4. Conclusion

A combined new global motion estimation algorithm is proposed. This algorithm consists of four-parameter global motion algorithm and robust estimator that is M-estimator for reducing outliers’ effects. Simulation results showed that the proposed global motion estimation algorithm significantly outperforms standard H.264/AVC codec and previous global motion estimation algorithm. The improvement in PSNR is about 0.2dB and the bit savings is about 1%. At specific sequence that has large object’s movement, the performance is much better than other algorithm. By reducing the global motion estimation’s computational complexity and accurate estimation, this algorithm can be used for real-time encoding of mobile device applications.

5. References

[1] Yeping Su, Ming-Ting Sun, Vincent Hsu, “Global Motion Estimation from Coarsely Sampled Motion Vector Field and the Applications”, Circuits and Systems for Video Technology, Vol. 15, Issue 2, pp. 232-242, 2005.

[2] Renan Coudray, Bernard Besserer, “Global motion estimation for MPEG-encoded streams”, Proc. of IEEE Int. Conference on Image Processing, Vol. 5, pp. 3411-3414, 2004.

Table 7: Evaluation of the ‘CIF’ Sequence

Sequence	QP	H.264/AVC JM 10.1		Previous Algorithm		Proposed Algorithm	
		Bit Rate (kbit/s)	PSNR (dB)	Bit Rate (kbit/s)	PSNR	Bit Rate (kbit/s)	PSNR (dB)
‘Mobile’ (CIF)	20	8411.89	40.21	8402.3	40.21	8385.39	40.23
	24	6310.03	36.6	6301.34	33.6	6278.46	36.64
	28	4678.94	33.25	4654.91	33.29	4646.51	33.31
	32	3371.83	29.63	3362.12	29.64	3350.56	29.71
	36	2372.23	26.17	2368.71	26.19	2351.45	26.28
	40	1649.6	23.02	1637.3	23.06	1629.64	23.28
‘Waterfall’ (CIF)	20	4315.91	40.35	4314.43	40.35	4311.83	40.35
	24	2798.5	36.84	2794.7	36.85	2793.67	36.85
	28	1811.86	33.76	1808.7	33.79	1806.88	33.81
	32	1219.15	30.77	1215.25	30.79	1201.25	30.8
	36	905.61	28.33	900.18	28.35	889.18	28.45
	40	752.87	26.5	752.81	26.33	750.83	26.71
‘Dragon’ (CIF)	20	4949.47	41.52	4867.06	41.52	4771.16	41.53
	24	3710.51	38.23	3676.23	38.26	3539.04	38.28
	28	2878.08	35.19	2842.04	35.2	2797.48	35.25
	32	2261.07	31.88	2248.51	31.96	2222.23	31.97
	36	1707.07	28.78	1705.87	28.92	1701.2	28.99
	40	1205.86	25.96	1201.43	26.11	1193.49	26.21

[3] David Corrigan, Anil Kokaram, Renan Coudray, Bernard Besserer, “Robust Global Motion Estimation from MPEG Streams with a Gradient based Refinement”, in Proc. Acoustics, Speech and Signal Processing (ICASSP 2006), Vol. 2, pp. 285-288, 2006.
[4] Aljoscha Smolic, Michael Hoeynck, Jens-Rainer Ohm, “Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications”, IEEE Int. Conference on Image Processing, Vol. 2, pp. 271-274, 2000.

Acknowledgement

This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2008-(C1090-0801-0011)).